

**METHOD AND SYSTEM FOR SCALING CONTROL  
IN 3D DISPLAYS (“Zoom Slider”)**

**CROSS REFERENCE TO OTHER APPLICATIONS**

- 5     This application claims the benefit of United States Provisional Patent Applications 60/505,345, 60/505,346 and 60/505,344, each filed on November 29, 2002, and all under common assignment herewith.

10    **FIELD OF THE INVENTION**

The present invention relates to the field of computer graphics, and more particularly to user interaction with computer-generated displays of three-dimensional (3D) data structures.

15    **BACKGROUND OF THE INVENTION**

When viewing an image on a computer or other electronically generated display, such as that of a photograph, a diagram or an X-ray, it is often necessary to examine one region in closer detail than is provided by the original resolution. As a result, most conventional image-viewing software has some type of scale-change controls, such as, e.g., a scale menu, magnify and shrink controls, or the like. Commonly, when a magnified image is larger than the available display window, the region being displayed is not the region of current interest, and a user must re-center the region of interest in the display.

- 25    Image viewing software may also support a directed magnification function whereby a user can specify a point in the original image which is used by the system as the

center of a magnified or enlarged image. Sometimes this center is set by the position of a mouse controlled cursor. In such contexts, clicking on the mouse causes the view to jump to an enlarged one with its center at the selected point, or “jump to zoom.”

- 5 A desirable feature in image viewing is smooth zooming. Unlike the jump to zoom function described above, in smooth zooming a point in the image stays fixed in the display and other points in the image move outwards from it. However, this is not supported in conventional image viewing software. Thus, users simply tolerate the need to manually slide the view vertically and horizontally after sizing jumps.

10

- In the viewing and manipulation of 3D, displays, the problems of magnification management become more acute for several reasons. First, in dealing with volumes, there is considerably more space to contend with. For example, a two-dimensional, or 2D image of an object that actually has four times the width and four times the height that fits in a window at a given resolution requires a user to look through sixteen
- 15 window-size regions to recover a given point of interest. However, a 3D image of the same object, similarly scaled to four times the width, height and depth of a viewing box, actually encompasses a volume sixty-four times as large as the viewing box.

- 20 Second, a 3D display generally includes more empty space than a 2D image. A 2D image can contain image content or detail at every point in the image. Since a 3D display must be looked at from a particular point in space, any detail between that spatial viewpoint and the object of interest obscures the view. As a result, empty

space may be required in 3D displays. When a 3D image is enlarged, however, this otherwise useful empty space tends to fill the display volume with such vast expanses of empty space that a user may have no clue whether to slide left or right, up or down, or forward or back to orient herself and find a particular area of interest.

5

Additionally, specifying a point in 3D presents user interfacing complexities. In what is termed a fully functional 3D interface, a user can move a stylus, pointer or other selector in three directions — horizontally across the display, vertically up and down the display, as well as along the direction into and out of the screen — and select a point. While this facilitates the “zoom from here” or close up mode, it is tedious to have to continually switch between overview and close up modes. In the more common mouse or other 2D interface, only two factors can be changed at a time. Thus, the interface can be set such that sideways motion of the interface produces a sideways motion of the cursor, and a vertical interface motion moves the cursor vertically, or, to adapt to 3D display control, the interface can be set (for example by depressing a mouse button) such that a sideways or vertical motion can be associated with the direction into/out of the screen (i.e., the depth dimension of a 3D display), or some fixed combination of these. However, there is no way that a two dimensional interface can control all three independent directions without added mode switching.

15  
20

A further complexity of 3D displays is that it is common (in order to see past features not currently of interest) to set a *crop box* outside which nothing is shown. This is effectively a smaller display box within the volume of space visible in the display window. A user must therefore be able to switch between moving the displayed data

— and with it the crop box — and moving the crop box across it. Distinct from the crop box, which is defined relative to the displayed model, is a *clipping box* which may exist in the same interface, and which typically has its size and location defined directly with reference to the *display* region, which, analogously to defining a

5 subwindow in a 2D interface (usually done with its sides parallel to those of the main window) defines a subvolume within the viewing box. Thus, no part of the model that would be rendered outside the clipping box is shown, which can be useful to limit the data displayed to an amount that can be handled at interactive speeds. While a user may shrink a crop box for similar reasons of performance, its primary use is to

10 pare away parts of the model for the sake of visibility. It moves with the model, and represents a choice of which part of the model to look at.

For general applications of the present invention it is important to distinguish the crop box from the *bounding box*, which also moves with the model but typically serves

15 different functions, such as checking quickly for collisions. If the bounding boxes of two objects do not overlap, neither do the objects, though if the objects do not fill their bounding boxes the collision of the boxes only means that collision of the objects must be checked in more detail. It is often useful to trigger selection or highlighting when a user-controlled cursor enters the bounding box of an object. In many

20 applications (such as in Computer Aided Design, or CAD) there may be a multiplicity of models, each with its own bounding box, but in such applications it is rare for the user to be able to adjust the bounding box, or for the bounding box to be coupled to the graphics by cropping — causing not to be rendered — parts of the model which lie outside it. Indeed, normally no point of the model does lie outside it. In a display

25 of the parts of an automobile, for example, the graphics functioning often requires that

each model have a bounding box that acts within the code, though not visible to or modifiable by the user, but it is rare to give each wheel, pipe, washer, *etc.*, an individual crop box by which part of it may be excluded from display. In certain applications of volume display, concerned with the rendering of rectangular blocks of 3D scan data, it can be useful to combine the functions of crop box and bounding box. In general however, they are distinguished.

The effects of these clip boxes and crop boxes can interact disadvantageously in regard to zoom functionality. With a small crop box (and thus no problem as regards display rendering speed or exceeding memory capacity) it can be disconcerting to have the box disappear when it passes out through the wall of a small dimensioned clipping box that was set to handle earlier performance problems. With a large crop box (including, for example, the result of zooming a smaller one) — enlarged with respect to display coordinates though not with respect to the model portion(s) it crops — the use of a clipping box may be essential for adequate performance. This interaction requires continual user attention to the logistics of viewing the model, and thus distracts from her ability to actually view the regions of interest in the actual model.

Within the objects of the invention is the provision of new techniques that simplify, automate, and optimize user interaction when scaling, navigating, observing and zooming such 2D and 3D images and models.

**SUMMARY OF THE INVENTION**

A system and method for controlling the scaling of a 3D computer model in a 3D display system include activating a zoom mode, selecting a model zoom point and setting a zoom scale factor are presented. In exemplary embodiments according to the present invention, a system, in response to the selected model zoom point and the set scale factor, can implements a zoom operation and automatically move a model zoom point from its original position towards an optimum viewing point. In exemplary embodiments according to the present invention, upon a user's activating a zoom mode, selecting a model zoom point and setting a zoom scale factor, a system can simultaneously move a model zoom point to an optimum viewing point. In preferred exemplary embodiments according to the present invention, a system can automatically identify a model zoom point by applying defined rules to visible points of a displayed model that lie in a central viewing area. If no such visible points are available the system can prompt a user to move the model until such points become available, or can select a model and a zoom point on that model by an automatic scheme.

**BRIEF DESCRIPTION OF THE DRAWINGS**

Fig. 1 depicts an exemplary system of coordinates used in describing a three dimensional display space according to an exemplary embodiment of the present invention;

Figs. 2A-2B illustrate the effects of scaling an exemplary 3D object from different points according to an exemplary embodiment of the present invention;

Fig. 3 illustrates the exemplary use of a crop box to display only a selected part of an object according to an exemplary embodiment of the present invention;

Figs. 4A-4B illustrate the exemplary effects of scaling a 3D object from points near to and distant from the boundary of a current display region or clipping box according to an exemplary embodiment of the present invention;

Fig. 5 illustrates various exemplary options for the selection of a Model Zoom Point  
5 with reference to a current crop box as opposed to with reference to a model according to an exemplary embodiment of the present invention;

Fig. 6 illustrates an exemplary Magnification Region defined by a planar Context Structure according to an exemplary embodiment of the present invention;

Figs. 7A-7D depict exemplary icons for a Model Zoom Point indicator according to  
10 an exemplary embodiment of the present invention;

Fig. 8 depicts an exemplary slider control object used for zoom control according to an exemplary embodiment of the present invention;

Fig. 9 illustrates an exemplary coordination of scaling with movement of the Model Zoom Point toward the Optimum Viewing Point according to a according to an  
15 exemplary embodiment of the present invention;

Fig. 10 depicts an exemplary process flow according to an exemplary embodiment of the present invention;

Fig. 11 is an exemplary modular software diagram according to an exemplary embodiment of the present invention; and

20 Figs. 12-18 depict an exemplary zooming in on an aneurysm according to an exemplary embodiment of the present invention.

**DETAILED DESCRIPTION OF THE INVENTION**

The present invention comprises a user-controlled visual control interface that allows a user to manipulate scaling (either by jumps or smooth expansion) with an easily-learned sense of what will happen when particular adjustments are made, and without becoming lost in data regions where nothing is displayed.

In an exemplary embodiment, an Optimum Viewing Point is fixed near the center of the screen, at a central depth in the display. When zooming control is active, a visual display of a cross or other icon around the zoom center marks this point. In an exemplary embodiment there is a larger contextual structure around the Optimum Viewing Point, indicating to a user a Magnification Region in which a Model Zoom Point will be selected. User controlled motion of the visible part of the model(s) in the display brings such model(s) into contact with the *z-axis* or the Magnification Region, and triggers the selection of a Model Zoom Point. If this fixed zoom center is not in line (from the user's viewpoint) with any point of the current crop box, the user is prompted to move the box together with its contents toward the center of the field of view. When a user begins to zoom model(s), the model space moves in the display such that the Model Zoom Point approaches the Optimum Viewing Point.

20

In another exemplary embodiment the system searches for and selects the model to be scaled and a candidate Model Zoom Point. This requires less effort from the user but correspondingly offers less detailed control.



The following terms of art are used throughout this application, and are defined here to facilitate the readability of the following description:

**3D data display** – a system capable of displaying images of 3D objects which present one or more cues as to the depth (distance from the viewer) of points, such as  
5 but not restricted to perspective, occlusion of one element by another, parallax, stereopsis, and focus. A preferred exemplary embodiment uses a computer monitor with shutter glasses that enable stereo depth perception, but the invention is equally applicable in the case of other display systems, such as, for example, a monitor without stereo, a head mounted display providing a screen for each eye, or a display  
10 screen emitting different views in different directions by means of prisms, aligned filters, holography or otherwise, as may be known in the art.

**Center of Scaling** - the point in a 3D model around which scaling takes place. Sometimes referred to herein as a “Model Zoom Point.”

**Display Space** - the 3D space whose origin is at the center of the display  
15 screen, used to orient points in the display screen. Points in Display space are denoted by co-ordinates  $(x, y, z)$ .

**Magnification Region** - a preferred 3D region displayed to a user by the system once zoom functionality is activated. Used by system to select a center of scaling point.

20 **Model Space** - the 3D space used to describe the model or models displayed by the 3D display system. Points in Model space are denoted by co-ordinates  $(u, v, w)$ , which related to display space by a co-ordinate transformation of the form

specified in Equation 1. Model space is fixed relative to the model; display space is fixed relative to the display device.

**Model Zoom Point** - the point on a model that remains fixed in a zoom operation, about which all other points are scaled.

5        **Optimum Viewing Point** - the center or near center point of a display screen at the apparent depth of the display screen. For simplicity of discussion we assume that this point is also chosen as the origin  $(0,0,0)$  of the display coordinates  $(x,y,z)$ , though this may be changed with trivial modifications to the algebra by one skilled in the art.

10

**Scaling** - multiplying the size of an object by a given number. A number greater than one effects a “zoom in” or magnification operation, while a number less than one effects a “zoom out” or reduction operation.

**Stereoscopic** - relating to a display system used to impart a three-dimensional  
15 effect by projecting two versions of a displayed scene or image from slightly different angles. There is a preferred viewing position relative to the display screen from where the stereoscopic effect is most correct, where the eye locations assumed in generating the separate visual signals coincide with actual locations of the user’s eyes. (At other positions the stereoscopic effect is equally strong, but the perceived form is  
20 distorted relative to the intended form.)

**Zoom** - see “Scaling.”

The methods of the present invention are implementable in any 3D data display system, such as, *e.g.*, a volume rendering system. As well, they may also be adapted and simplified for display of two-dimensional (2D) data, in ways evident to those skilled in the art. In general, a volume rendering system allows for the visualization  
5 of volumetric data. Volumetric data is digitized data obtained from some process or application, such as MR and CT scanners, ultrasound machines, seismic acquisition devices, high energy industrial CT scanners, radar and sonar systems, and other types of data input sources. One of the advantages of volume rendering, as opposed to surface rendering, is that it allows for the visualization of the insides of objects.

10

One type of such 3D display system is what is referred to herein as a fully functional 3D display environment (such as, *e.g.*, that of the Dextroscope™ system of Volume Interactions Pte Ltd of Singapore, the assignee of the present application). Such systems allow for three-dimensional interactivity with the display. In such systems a  
15 user generally holds in one hand, or in each hand, a device whose position is sensed by a computer or other data processing device. As well, the computer monitors the status of at least one control input, such as, *e.g.*, a button, which the user may click, hold down, or release, etc. Such devices may not be directly visible to the user, being hidden by a mirror; rather, in such exemplary systems, the user sees a virtual tool (a  
20 computer generated image drawn according to the needs of the application) co-located with the sensed device. In such exemplary systems the locational identity of the user's neuromuscular sense of the position of the held device, with the user's visual sense of the position of the virtual tool is an interactive advantage.

Fig. 1 depicts an exemplary co-ordinate system for 3D displays. With reference thereto there is a plane 103, which represents the apparent physical display window at a preferred central depth in the display. Display window 103 is generally a computer screen, which may be moved to a different apparent position by means of lenses or  
5 mirrors. Alternatively, in head-mounted displays the referent of plane 103 is a pair of screens occupying similar apparent positions relative to the user's left and right eyes, via a lens or lenses that allow comfortable focus of the eyes. In the case of a stereoscopic display, the preferred central depth is at or near the distance at which the user sees the physical surface of the monitor, in some cases via a mirror. Around this  
10 distance the two depth cues of stereopsis and eye accommodation are most in agreement, thus leading to greater comfort in viewing.

At or near the center of display window 103 is an origin 102 of an orthogonal co-ordinate system having axes 107, 108 and 109 respectively. As well, a schematic  
15 head and eye 115 are shown, representing the point from which the display is viewed by a user. For ease of illustration, the following conventions will be used given the location of such user 115: the  $x$ -axis, or horizontal axis, is 107, the  $y$ -axis, or vertical axis, is 108, and the  $z$ -axis, or depth axis, is 109. Positive directions along these axes are designated as rightward, upward and toward the user, respectively. 'Greater  
20 depth' thus refers to a greater value of  $(-z)$ . The origin 102 as so defined has display space around it on all sides, rather than being near a boundary of the display device. Furthermore, for many users the agreement of optical accommodation with depth cues such as stereopsis (and in certain exemplary display systems parallax) causes an object at this depth to be most comfortably examined. Such an origin,  
25 therefore, is termed the Optimum Viewing Point.

It is understood that real world considerations of ergonomics and perceptual psychology may lead to a variant choice of origin, for a particular application and hardware configuration, that is not precisely centered in the physical display screen or  
5 not at the apparent depth of the physical display screen.

A 3D display system generally displays at least two kinds of objects: *control interfaces* such as buttons, sliders, etc., used to control system behavior, which typically retain their apparent size and position unless separately moved by the user,  
10 and the actual *3D models* (generated from some external process or application such as, *e.g.*, computerized tomography, magnetic resonance, seismography, or other sensing modalities), geometric structures defined by points, lines and polygons or implicit equations such as  $f(x, y, z) = 0$ , where  $f$  is some suitably defined function. The 3D models are equipped with attributes, such as, for example, colors as well as  
15 other necessary data, such as, for example, normal vectors, specularity, and transparency, as may be required or desirable to enable the system to render them visually in ways that have positions and orientations in a shared *model space*.

Model space is said to have coordinates  $(u, v, w)$  whose relation to the display region  
20 coordinates (as specified by axes 107, 108, and 109 in Fig. 1) is given by the matrix relation

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_1^1 & a_1^2 & a_1^3 \\ a_2^1 & a_2^2 & a_2^3 \\ a_3^1 & a_3^2 & a_3^3 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} + \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (1)$$

or some equivalent such as, for example, a 4x4 matrix formulation of the same affine (straight-line-preserving) transformation. As can be seen from Equation 1, the origin of the model space, where  $(u, v, w) = (0, 0, 0)$ , maps to the display space point

- 5  $(x, y, z) = (X, Y, Z)$ . Changing the position  $(X, Y, Z)$  to which  $(u, v, w) = (0, 0, 0)$  is mapped to thus *translates* the model space and all models in it within the display space.

- Zoom or scaling functionality operates as follows. Multiplying the matrix  $[a_i^j]$  of
- 10 Equation 1 by a number  $\lambda$  shrinks or magnifies the appearance of the objects and the distances between them by that factor, keeping any visible element with  $(u, v, w) = (0, 0, 0)$  at the position  $(X, Y, Z)$  while other points move either toward or away from the point  $(X, Y, Z)$  as the distances are scaled. Combining a change of  $(X, Y, Z)$  with a scale change allows a point other than  $(u, v, w) = (0, 0, 0)$  in model
- 15 space to retain a constant position in display coordinates  $(x, y, z)$ , in either a single step or a succession of small steps that give a sense of continuous change.

- Figs. 2A and 2B show two choices 201 of such an unmoving or fixed point denoted by a “+” icon, with the different effects of scaling an object 202 to be three times
- 20 larger 203 along all three axes (and hence in all directions). In Fig. 2A Optimal

Viewing Point 201 is chosen to remain fixed. Thus, all points on the expanded object 203 remain centered about that point. In Fig. 2B a point somewhat translated from (0,0,0) was chosen as the center of scaling. Thus in Fig. 2B the center of expanded object 203 has moved within the display space.

5

If the point  $(U, V, W)$  is to appear fixed, we replace the above equation (1) by

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \lambda \begin{bmatrix} a_1^1 & a_1^2 & a_1^3 \\ a_2^1 & a_2^2 & a_2^3 \\ a_3^1 & a_3^2 & a_3^3 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} + \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} \quad (2)$$

where  $\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \left( (1 - \lambda) \begin{bmatrix} a_1^1 & a_1^2 & a_1^3 \\ a_2^1 & a_2^2 & a_2^3 \\ a_3^1 & a_3^2 & a_3^3 \end{bmatrix} \begin{bmatrix} U \\ V \\ W \end{bmatrix} + \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \right)$

The point  $(U, V, W)$  is termed the *model center* of the scaling, and the point to which it corresponds under the transformations to both the left and the right of Equation 3

$$\lambda \begin{bmatrix} a_1^1 & a_1^2 & a_1^3 \\ a_2^1 & a_2^2 & a_2^3 \\ a_3^1 & a_3^2 & a_3^3 \end{bmatrix} \begin{bmatrix} U \\ V \\ W \end{bmatrix} + \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} a_1^1 & a_1^2 & a_1^3 \\ a_2^1 & a_2^2 & a_2^3 \\ a_3^1 & a_3^2 & a_3^3 \end{bmatrix} \begin{bmatrix} U \\ V \\ W \end{bmatrix} + \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (3)$$

is known as the *display center* of the scaling. Note that the two sides of (3) are not identically equal for all  $(U, V, W)$ : the truth of the equality for a particular  $(U, V, W)$ , necessarily unique if  $\lambda$  is not equal to one, precisely characterizes that  $(U, V, W)$  as the display center of the scaling. In a one-dimensional analogy,

15  $\lambda aU + X' = aU + X$  precisely if  $U = (X' - X)/(a - \lambda a)$ . A model center of scaling is also known as a “zoom point” or “Model Zoom Point.” The correspondence between display space coordinates and model space coordinates may also be modified by

rotation, reflection and other geometric operations by multiplying the matrix  $[a_i^j]$  by other appropriate matrices as may be known in the art. As well, the positions of models within the model space (and hence relative to each other) can be modifiable in an application. Thus, while the present description of the invention addresses

5 primarily the common scaling of objects in a shared model space, the extension to the case of one or more such objects, each in a separate model space that is itself related to the main model space will be evident to one skilled in the art.

The use of a user controlled crop box is distinct from the inherent clipping that is

10 caused by the finite nature of the display apparatus. As a physical limitation of the display system, no part of a model can be shown outside the region that the system is configured to display to the user. However, this region can be further user-restricted by the use of a clipping box as described above. Figs. 4 illustrate the effect of a zoom point near the boundary of the available display region (or current clipping box). In

15 Fig. 4A zoom point 401 is chosen. Point 401, being centered with respect to the crop box boundary 450, causes minimal loss of the enlarged object 405 from view. On the other hand, in Fig. 4B, where zoom point 412, located near the left boundary of the crop box 450, is chosen, upon magnification of the object 411 to the enlarged object 413, large portions are lost from view. This choice of zoom point makes more of

20 model 411 move out of view and become undisplayable than occurs for the same model 410 using model zoom point 401 which is in a more central location.

However, it is often the case that a user desires to zoom an object using a model zoom point that *is* near a boundary (either crop box or viewing box). The reduction of the resulting inconvenience to the user is a major aim of the present invention. (Note that



a similar effect occurs if the model zoom point is near the surface of the current crop box, but since the user frequently manipulates the crop box this is less often a problem).

- 5 As noted above, the present invention may be implemented in various display system environments. For illustration purposes its implementation is described herein in two such exemplary environments. An exemplary preferred embodiment of the invention is in a Dextroscope™-like environment, but the invention is understood to be fully capable of implementation using, for example, a standard mouse-equipped computer,
- 10 or an interface using hardware not otherwise mentioned here such as a joystick or trackball, or using a head-mounted display, or any functional equivalent. Although for illustrative purposes the range of options is described with reference to a Dextroscope™-like environment and a standard mouse, adaptations to other equipment will be clear to one skilled in the art.
- 15 Sections 1-4 below describe the successive steps of user interaction with a display according to the present invention.

#### **1. Activate Zoom Mode**

- Initially a user signals that the system should enter a state ('zoom mode') in which
- 20 signals are interpreted by the system as directed to the control of magnification rather than other features. This may be done through, for example, a voice interface (speaking a command such as 'engage zoom mode' which is recognized by the system), by clicking on a 2D button with the mouse or on a 2D button with a Dextroscope™-type stylus, by touching a button, or in a preferred exemplary

embodiment, by merely touching (as opposed to clicking) a zoom slider interface object as described below (in connection with Fig. 8).

## **2. Select Model Zoom Point**

5 Next, a user selects a point in the model space, termed Model Zoom Point, around which it is desired to see more detail by scaling around it. Examples of such a Model Zoom Point are the points 201 shown in Figs. 2. This selection may be done in a number of ways. For example, in a mouse interface, the user may click on a point of the screen, and the Model Zoom Point selected will be the nearest visible (non-  
10 transparent) point on the model that is in line with that point from the user's viewpoint. (In the case of a stereo display, where the user has two viewpoints, the system may select one eye for this calculation.) Alternatively, in a more exact but simultaneously more demanding interface, a moving "currently selected" point can be displayed that the user may select with some input interaction, such as, e.g., an  
15 additional click. In a Dextroscope™-like interface, for example, the user may click on a three dimensional point (on or off of the visible surface of the displayed object) which then becomes the Model Zoom Point. Other such selection means may be utilized as may be known in the art.

20 In the first method described below for the selection of a Model Zoom Point, selection is integrated with the placement of the model in a convenient place for zooming within the display space. Just as with a 2D image, a zoom point near the boundary of the display region makes points near it disappear over that boundary more quickly than a central point does, as was illustrated in connection with Figs. 4. Moreover, in a

stereoscopic display there is a most comfortable viewing depth, which coincides with the real or apparent distance from the user's eyes of the physical display screen.

## 2.1 First Selection Method

5

### A. Crop Box Enabled

Thus, in an exemplary embodiment, the system uses a centering method for assisting a user in selecting the Model Zoom Point. The system examines the  $z$ -axis of the display coordinate system  $(x, y, z)$  to find the nearest point in display space  $(0, 0, Z_0)$  at which the current display includes a visible point of some model in model space, in the current position of model space relative to the display coordinates. If such a point exists, by being visible it is necessarily inside the current crop box (if such a crop box is available and enabled, as is typical for, *e.g.*, volume imaging but less so for, *e.g.*, rendering a complex machine design or virtual film set.). If no such point can exist for the reason that the  $z$ -axis does not pass through the crop box given the current position of the crop box, the reader is then prompted to move the crop box (with its contents, so that the change is of the transformation type quantified in Equation (1), as discussed above) until the crop box does meet the  $z$ -axis. This may be accomplished, for example, in a Dextroscope™-like display system by the user grasping the box with the workpiece hand, moving the sensor device, the tool (visible or logical) attached to the sensor device, and the box in coordination with the tool, until the box is sufficiently centered in the display for such a passing through of the  $z$ -axis to occur. Alternatively, in display environments using a standard 2D mouse interface, for example, the screen position of the box may be dragged across the display in a

standard 'drag and drop' action, since the  $z$ -component of the motion is not impacted in such an operation, and the object may be maintained through this step at constant  $z$ .

#### **B. Crop Box But No Point Visible on $z$ -axis**

5 If a crop box is currently enabled, but the  $z$ -axis encounters no visible point of a model thereon, in an exemplary embodiment the system may determine  $Z_0$  by a default rule involving box geometry as is illustrated in Fig. 5. For example, it may define  $(0, 0, Z_0)$  as (a) the point nearest the user 501 (in Fig. 5 the user 500 views from the far left of the Figure) at which the  $z$ -axis 510 meets the crop box 520; (b) the  
10 point farthest from the user 502 at which the  $z$ -axis meets the box; (c) the mid-point 503 of the latter two points 501, 502; (d) the point on the  $z$ -axis nearest the centroid of the box, (e) the  $z$ -value of the  $(x, y, z)$  position of the centroid of the box; or (f) such other rules as may be desirable or useful in a given design context. Alternatively, it may determine  $Z_0$  by a default rule involving the crop box contents. For example, it  
15 may, as above, set  $Z_0$  at the  $z$  value of the  $(x, y, z)$  position of the nearest point to the  $z$ -axis at which a visible point on a model exists, or it may define  $Z_0$  as the  $z$  value of the  $(x, y, z)$  position of the centroid of the points in the box that are currently treated as visible rather than transparent. Numerous other alternatives as may be known in the art may be implemented in various alternative exemplary embodiments.

20

#### **C. No Crop Box Enabled**

If there is no crop box or equivalent functionality, the system may, in an exemplary embodiment, set the Model Zoom Point to be the center of the currently selected

model, the origin of internal model coordinates (distinct from general model space coordinates  $(u, v, w)$ ) which may or may not coincide with such a center, the center of the bounding box of the current model, the Optimum Viewing Point or the origin  $(0, 0, 0)$ . (If Optimum Viewing Point is chosen, however, the description below of the movement of the center of scaling becomes moot.) Alternatively, it may use for  $Z_0$  the  $z$  value of the  $(x, y, z)$  position of the nearest point to the  $z$ -axis at which a visible point on a model exists (though such a search could be overly computationally expensive). Numerous other alternative choices may be implemented in other exemplary embodiments as may be desired or appropriate in given design contexts.

## 2.2 Second Selection Method

In a preferred exemplary embodiment of the invention, a second centering method for selecting the Model Zoom Point can be utilized, as illustrated in Fig. 6. With reference to Fig. 6, this method utilizes the concept of a Magnification Region 603. A Magnification Region is a central region (fixed by the system or adjusted by the user, and made visible to the user) of display space, within which a visible point of the model or its crop box may be selected. In an exemplary embodiment, this region is shown by displaying a translucent Context Plane 602. As its name implies, a Context Plane covers much of the screen, leaving a hole 601 (circular in the example depicted in Fig. 6, but other shapes, such as, for example, square, rectangle, ellipse, hexagon, etc. may equally be used) around the center of the screen. More precisely it is preferred that the Context Plane be rendered such that its hole's centroid is the Model Zoom Point of the display space, since 'centered in the hole' is more easily apparent to the user than other relationships. However, other such context structures and relations will be apparent to one skilled in the art. Such a Context Plane 602 may

preferably be drawn after all other rendering and with the depth buffer of the graphics rendering system turned off, so that the colors of images at all apparent depths are modified by it to thus highlight the hole. Such color modification may, for example, comprise blending all pixels with gray, so that the modified parts are de-emphasized, and the parts within the hole 601 are emphasized.

In the case of a stereo display the plane is physically rendered for each eye, so that it has an apparent depth. If this depth is set at that at which the user perceives the display screen to be (for example, through mirrors, lenses or other devices, such that it need not be the physical location of the display surface), it is rendered identically to each eye. The apparent depth of the display screen is often most preferred for detailed examination, but other depths may be used as well. Parenthetically, it is noted that a structure rendered translucently without reference to the depth buffer after other stereo elements have been rendered can be of use for any 3D-located feature of a display, not only as an icon marking a Model Zoom Point. Moreover, since its perceived depth is less certain than a structure rendered with the added depth cue of occlusion, in a preferred exemplary embodiment such translucency is utilized only for a context structure, such as the Context Plane 602, to call attention to an opaquely rendered point marker.

20

However, in a stereo environment with additional depth cues, such as, for example, parallax (i.e., the change of appearance in response to motions of the user's head and eyes, which are tracked by the system), the location of an object rendered in such translucent manner may appear very definite to the user, making this a useful

technique for placing one object within another. It permits the user's visual system to construct (perceive) a consistent model of what is being seen. Thus, suppose that from a given viewpoint two opaque objects are rendered, the first of which geometrically occludes the second by having parts which block lines of sight to parts of the second, but the second is rendered after the first, opaquely replacing it in the display. The user is faced with conflicting depth cues. Stereopsis (plus parallax and perspective if available) indicate the second as more distant, while occlusion indicates that it is nearer. However, if the second object is translucently rendered, the visual system can resolve the conflict by perceiving that the second is visible through the first, as though the first were translucent to light from (and only from) the second object. While this is not common with physical, non-virtual objects, the mental transfer of transparency to an object that was in fact opaquely rendered appears to be automatic and comfortable to users. Such technique is thus referred to as *apparent transferred translucency*.

With reference again to Fig. 6, the hole 601 in the Context Plane 602 defines the Magnification Region 603 (or, more precisely, the cross-section of the Magnification Region at the z-value of the Context Plane). In a monoscopic display environment this region is the half-cone (or analogous geometric descriptor for a non-circular shape used for the hole 601) consisting of all points lying on straight lines that begin at the viewpoint of a user 610 (element 610 is a stylized circular "face" with an embedded eye, intended to schematically denote a user's viewpoint) and that pass through the hole 601 in the Context Plane 602. In a stereoscopic display the Magnification Region may be defined in a variety of ways, such as, for example, the corresponding cone for the right eye's viewpoint, the cone for the left eye's viewpoint, the set of points that are in the cones for both eyes' viewpoints (i.e., the

intersection of the two cones), or as the set of points that are in the cone for either eye's viewpoint (i.e., the union of the two cones), or any contextually reasonable alternative. Commonly, such regions are further truncated by a near and a far plane, such that points respectively nearer to or farther from the user are not included.

5

In an exemplary embodiment which uses a Magnification Region to select a Model Zoom Point, after displaying the Magnification Region the system determines whether there is a visible point of the model (or alternatively of the crop box) in the Magnification Region. If there is not, the user is then prompted by the system to  
10 move the model.

If there is such a point, the system selects the nearest such point as a Model Zoom Point. There are various possibilities for defining "nearest" in this context. For example, the system may select the nearest visible point of the model (or  
15 alternatively, of the crop box) within the Magnification Region which lies at a greater depth than the Context Structure. (In most 3D display environments it is unnecessary to examine all of the points in the model to find this point, inasmuch as a depth buffer generally records the depth of the nearest visible point along each line of sight). Or, for example, the system may select the nearest such point, which lies on the viewer's  
20 line of sight through the center of the Context Structure (and this choice may also be subject to the condition that the point be at a greater depth than the Context Structure). Finally, for example, the system may select a point by minimizing the sum of squared depth beyond the Context Structure and squared distance from the viewer's line of sight through the center of the Context Structure, multiplied by some chosen



coefficients to emphasize depth proximity or centrality respectively. Many alternative selection rules can be implemented as desired in various embodiments of the invention.

5     **2.3 Third Selection Method**

It is noted that in the first and second methods above, the choice of which model (supposing there are several in a model space) the Model Zoom Point is to lie on is implicit in the concern with proximity to the z-axis. The user wishing to enlarge one model rather than another may simply arrange that it meets the z-axis or the

10    Magnification Region and is the nearest model to do so. However, in a complex scene such as is common in CAD, the user may not wish to move (for example) the position of a whole engine in order to temporarily examine a gearbox in more detail. The system may, in an exemplary embodiment, thus optionally provide for the user to identify one or more of the displayed models (by, for example, clicking on their

15    images with a 3D stylus or 2D mouse, by calling out their names, or otherwise), and may then optionally select the center of a single or first-chosen or last-chosen model, the centroid of all chosen models, or other point selections as may be known in the art. Alternatively, the system may prompt the user to select a Model Zoom Point by whatever means of 3D point selection is standard in the application, or may offer the

20    opportunity to select such a point in replacement of the automatic selection just described.

In a preferred exemplary embodiment of the invention, these selections of model and Model Zoom Point can be, for example, automated and integrated, as described in the

25    following pseudocode. It is noted that the system begins by testing whether the user

has aligned a model with the z-axis or Magnification Region as described above, but if she has not the preferred exemplary embodiment defaults to an automatic scheme rather than prompting her to do so. In what follows the convention that text following a // is a comment descriptive of the active code is followed, though the 'code' itself is

5 .simplified for clarity. "Widget" refers to some interactive display object.

```

class ScalingControl {
// defines the Control Widget for Scaling: typically a slider

public: // functions callable by other parts of the program, described more fully below

    bool Contain (Point); // test for containing a given point
    void Update (); // modify graphics according to the scale value read from the widget
    void Active (); // use the fact that the user touches the widget (without depressing
15         // a button, to drag and change its scale value) to engage zoom mode

    void Update_Model_Zoom_Point (); // modify Model zoom Point according to the state
                                     // of the widget

20 void Render_Context_Plane (); // add Context Plane, appropriately placed, to display
    void Render_Model_Zoom_Point (); // add Zoom Point icon to display private:
    Point mModelZoomPoint; // data accessible only to the functions of ScalingControl
};

// 25 the following pseudodefintions describe the functions performed
// by the functions introduced above.

bool ScalingControl::Contain (Point p)
{
30 // Return true if p is inside the scaling control widget as positioned in display space.
    // Otherwise, return false.
}

void ScalingControl::Update ()
// 35 This function is called when the scaling control is being focused
// but not yet triggered.
{
    Update_Model_Zoom_Point ();
    Render_Context_Plane ();
40 Render_Model_Zoom_Point ();
}

void ScalingControl::Active ()
// This function is called when the scaling control is triggered
// 45 And active.
{
    Move_Object_To_Optimum_Viewing_Point ();
    model.size = Get_Scale_Factor ();
    if (model.size > size_threshold)
50 Show_Clipping_Box ();
    else
        Hide_Clipping_Box ();
    Draw_Model_Zoom_Point ();
}

```

```

}
bool ScalingControl::Update_Model_Zoom_Point ()
{
    // Search for a Model Zoom Point using four methods:
    // Method 1 - select user-nearest point visible (if any along) z-axis.
    // if none visible, try
    // Method 2 - select user-nearest point visible along the line from user's
    // viewpoint to the Center of All Visible Objects
    // if none visible, try
    // Method 3 - select user-nearest point visible along the line from user's
    // viewpoint to the Center of Each Visible Object (sorted according to the
    // centers' distance to the Optimum Viewing Point)
    // if none visible, use
    // Method 4 - Use the Center of the Object nearest to the Optimum Viewing Point
    if (Clipping_Box_Is_Enabled ())
    {
        if (Model_Ray_Intersection (CENTER_OF_THE_SCREEN, &mModelZoomPoint))
            return true;
    }
    return false;
}
else
{
    if (Model_Ray_Intersection (CENTER_OF_THE_SCREEN, &mModelZoomPoint))
        return true;
    if (Model_Ray_Intersection (CENTER_OF_ALL_OBJECTS, &mModelZoomPoint))
        return true;
    if (Model_Ray_Intersection (CENTER_OF_EACH_OBJECT, &mModelZoomPoint))
        return true;
    if (Model_Ray_Intersection (CENTER_OF_NEAREST_OBJECT, &mModelZoomPoint))
        return true;
    return false; // if none of the above tests is passed.
}
}
35
void ScalingControl::Move_Object_To_Optimum_Viewing_Point ()
{
    // Move the Object and the Model Zoom Point (mModelZoomPoint) to
    // the Optimum Viewing Point.
}
40

void ScalingControl::Render_Context_Plane ()
{
    // Disable Depth Buffer checking.
    // Render a semi-transparent plane with a hole inside. The center
    // of the hole should be mModelZoomPoint.
}

void ScalingControl::Render_Model_Zoom_Point ()
{
    // Render a 3D crosshair with mModelZoomPoint as the position.
}

```

The following Program Entry Point pseudocode illustrates the way in which the above exemplary functionality can be, for example, called by a functioning application.

```

void main ()
{
    // Set up variables and states, create objects
    Initialization ();

    Tool tool; // Create one 3D tool
    ScalingControl scalingControl; // Create one control widget
10    while (true)
    {
        Render_Model ();

15    if (scalingControl.Contain (tool.GetPosition ()) // If 3D tool is inside control widget
        {
            if (tool.IsButtonPressed ()) // If the 3D tool button is pressed
                scalingControl.Active (); // Bring Model Zoom Point to Optimum Viewing Point
            else
20        scalingControl.Update (); // Update value of Model Zoom Point
        }

        UpdateSystem(); // Update all display and system variables
    }
} 25

```

## 2.4 Optional Modification By User

It is noted that in various exemplary embodiments utilizing a Magnification Region the Model Zoom Point is automatically selected, according to defined rules, as

30 described above. The user's control over this process consists of what she places within the Magnification Box, or what part of what model she was viewing prior to activating the zoom function, or what model is nearest the Optimum Viewing Point. Such automatic selection of the Model Zoom Point relieves the user of 'logistical' concerns, thus allowing her to focus on the work at hand. However, an exemplary

35 embodiment may allow the user at any point to invoke the procedure described above, at the beginning of the current Section 2, 'Selection Of Model Zoom Point', to choose a different point and override the automatic selection.

### 3. Display Of Model Zoom Point

With reference to Fig. 7, after selection of a Model Zoom Point, the system displays an icon, such as for example, a cross, at the Model Zoom Point. In the exemplary embodiment depicted in Fig. 7B, an everted cross 700 of four triangles 710, each  
5 pointing inward toward the Model Zoom Point 711 is utilized. Any desirable icon form or design may be substituted, such as, for example, other flat patterns of polygons 715, a flat cross 705, or a three-dimensional surface structure 720 as depicted in Figs. 7C, 7A and 7D, respectively. In an exemplary embodiment the icon is drawn at the depth of the Model Zoom Point, rather than as simply a marker on the  
10 display screen. Thus, in a stereo display it is drawn such that a user whose visual system is capable of stereo fusion will perceive it to be at the intended depth, and that it is drawn as an opaque object capable of hiding objects behind it and of being hidden by objects between it and the user's eye or eyes. These different depth cues strengthen the user's sense of its three-dimensional location. However, since the use  
15 of opacity allows all or part of the icon to be obscured by the object displayed, in a preferred exemplary embodiment the displayed Context Plane is moved to lie at the same depth, with its hole centered on the Model Zoom Point. In an alternate exemplary embodiment, the icon can be rendered with apparent transferred translucency, as discussed above. Alternative exemplary implementations of the  
20 invention could function without context cues in locating the Model Zoom Point, or could use other cues such as, for example, a system of axes or other lines through it, with or without apparent transferred translucency.

#### 4. Zoom Control

Once a Model Zoom Point has been selected, zooming is enabled. Zooming can be controlled in a variety of ways, such as, for example, (1) voice control (with the system recognizing commands such as (a) “Larger” or “Smaller” and responding with a step change in size, (b) “Reset” and restoring the previous size, (c) “Quit zoom mode”, etc.); or (2) step response to key strokes, mouse or tool clicks on an icon, or clicks on a particular sensor button, etc. For example, while in zoom mode a middle mouse button click might automatically mean “Larger” while a right click is interpreted as “Smaller.” In a preferred exemplary embodiment a slider such as depicted in Fig. 8 is utilized. Such a slider may also be used as a zoom mode trigger, when, for example, a user uses a stylus to touch its body 801 without clicking. In such an exemplary embodiment, when a user places the point of the 3D stylus in or near the slider bead 802 and holds down the sensor button while moving the sensor, this moves the stylus and drags the slider bead 802 along the slider bar 801. The distance moved is mapped by the system to a magnification factor, by an appropriate algorithm. In an exemplary embodiment the algorithm assigns the minimum allowed value for the zoom factor  $\lambda$  to the left end of the slider 810, the maximum allowed value to the right end of the slider 811, and interpolates linearly between such assignments. Alternative exemplary embodiments include an exponential or logarithmic relation, or a function defined by assigning  $\lambda$  values at certain positions for the slider bead 802 and interpolating in a piecewise linear, polynomial or rational B-spline manner between them, or a variety of other options as may be known in the art. Other exemplary alternatives for the control of  $\lambda$  include the use of a standard mouse-controlled slider, a control wheel dragged around by a Dextroscope™-like system stylus, a physical scrolling wheel such as is known in certain mouse designs,

etc. In order to facilitate both magnification as well as reduction operations, the value range of  $\lambda$  may run from some minimum value less than unity (maximum reduction factor) to some maximum value greater than unity (maximum magnification factor).

The range need not be symmetric about unity, however, inasmuch as some

5   embodiments utilize magnification to a greater extent than reduction, and vice versa.

## **5.       Automatic Translation Of Model Zoom Point**

In coordination with the zooming function, in an exemplary embodiment the model space is moved in a manner intended to add to a user's comfort and convenience in

10   examining a model, while avoiding the perceptive disjunction that would follow from

large shifts. If the position in display coordinates  $(x, y, z)$  of the Model Zoom Point

is  $(x_0, y_0, z_0)$  before zooming begins, depicted as the point 901 in Fig. 9, the value of

$\lambda$  is used to calculate to a translation towards the Optimum Viewing Point 903 in

such a way that the unzoomed starting value  $\lambda = 1$  in effect at the start of zooming

15   connects to a zero translation (leaving the model space unmoved), while a large value

connects to a translation by a vector equal or near to  $(-x_0, -y_0, -z_0)$ , moving the

Model Zoom Point 901 via intermediate positions such as 902 to a display point

$(x', y', z')$  at or near the Optimum Viewing Point 903. This will typically produce the

most comfortable location of the zoomed model for detailed viewing and

20   manipulation, while not losing the original layout of the larger context. In a wholly complementary manner, upon reducing the zoom factor the Model Zoom Point moves back toward its original location 901.

In particular, if the system is adjusted to allow a maximum scale value of  $\lambda = \lambda_{\max}$ , for  $\lambda > 1$  we may in a particular implementation define  $t = \frac{\lambda - 1}{\lambda_{\max} - 1}$  and translate the display of the model space by  $(-tx_0, -ty_0, -tz_0)$ . Intermediate scaling of the model 912 is thus associated with an intermediate position 902 for the Model Zoom Point, and

5 maximum scaling of the model 913 coincides with translation of the display position of the Model Zoom Point 901 exactly to the Optimum Viewing Point 903. For  $\lambda < 1$  we may use the same formula, resulting in a movement away from the Optimum Viewing Point as the display size diminishes, or alternatively, for a minimum scale value of  $\lambda_{\min}$  replace it by  $t = \frac{\lambda - 1}{\lambda_{\min} - 1}$  so that for the extreme case the translation by

10  $(-tx_0, -ty_0, -tz_0)$  again moves the Model Zoom Point exactly to the Optimum Viewing Point. These formulae may be replaced by many others evident to those skilled in the art, such as exponential or polynomial functions, subject only to the condition that on each side (separately considered) of  $\lambda = 1$  the change in  $t$  should be monotonic (always increasing, or always decreasing, for an increase in  $\lambda$ ), so that the

15 model does not advance and retreat in ways surprising to the user. Two particular functions  $t(\lambda)$  that may be included in this framework are noteworthy. One extreme case sets  $t(\lambda) = 0$  for all values of  $\lambda = 1$ , so that the model space does not move at all apart from the effects of scaling, and the Model Zoom Point remains fixed in display space. The other extreme (used in a preferred exemplary embodiment) sets  $t(\lambda) = 1$

20 for all values of  $\lambda \neq 1$ , so that the Model Zoom Point moves immediately to the Optimum Viewing Point, before user-elected zoom values come into play. For continuity of the user perception reasons, in a preferred exemplary embodiment  $t$  is allowed to move continuously (that is, through a sequence of changes small enough to



give an impression of smooth motion) to the value  $t = 1$ ; such change to occur either (a) immediately upon the selection of a Model Zoom Point, or (b) when the user begins to modify  $\lambda$  by whatever method is selected.

5     **6.       Automatic Activation Of Measures To Preserve Performance**

Since a large value of  $\lambda$  increases the displayed size of the crop box, and hence the load on the graphics rendering hardware, performance may degrade (sometimes abruptly) in the course of zooming, whether or not coupled to motion of a selected model point such as the translation of the Model Zoom Point described above.

10    Therefore, in a preferred exemplary embodiment a load estimation function is attached to the crop box (such as, for example, a computation of the volume it currently occupies in the display space, which can be multiplied or otherwise modified by one or more factors specifying (a) the density of the rendering used: (b) the spacing of 3D texture slices; (c) the spacing of sample points within a 3D texture  
15    slice, where permitted by the hardware; (d) the spacing of the rays used (in embodiments utilizing a ray-casting rendering system); or such other quantities that can modify the quality and speed of rendering in a given exemplary system). When such load estimate reaches a threshold value (generally set by experiment with the particular hardware or derived from analysis of its specifications) at which there is a  
20    significant risk of performance degradation, the system automatically activates the clipping box at a default or user-specified size and position values, without requiring any affirmative user intervention. Alternatively, factors such as (a) to (d) just described may be automatically modified to reduce the load. Conversely, if the current load is below a threshold (typically set lower than the threshold above), the  
25    system, in an exemplary embodiment, may enlarge or remove the clipping box, or so

modify the factors such as (a) to (d) as to improve the quality of the rendering while increasing the load within supportable limits.

## **7. Exemplary Process Flow**

- 5 Figure 10 is a flowchart depicting process flow in an exemplary preferred embodiment of the invention. With reference to Figure 10, the following events occur. At 1001, process flow begins when a user indicates to the system that she desires to scale an object in a model. The user may indicate such directive by, for example, moving an input device such that the tip of a displayed virtual tool is inside
- 10 the bead 802 of a displayed control object, such as, for example, the zoom slider 801 depicted in Figure 8. As described above, numerous alternative embodiments may have numerous alternative methods of signaling the system that a zoom or scaling function is desired.
- 15 At 1002, the system determines whether a visible object is inside the Magnification Region. If there is such a visible object, process flow moves to 1004 and selects the center of magnification/reduction as in the First or Second Method above. If there is no such object, then according to the method chosen from those described above the system shall, in an exemplary preferred embodiment, enter an automatic selection
- 20 process such as that illustrated by the pseudocode above. Alternatively, as shown in this diagram 1003, it prompts the user to move the object until the determination gives a positive result, upon which it can proceed to 1004.

As described above, either these or numerous alternative methods support the selection of a Model Zoom Point, depending upon the type of display environment used in a particular embodiment of the invention, as well as whether a crop box and/or Magnification Region is utilized, etc.

5

Once the Model Zoom Point is selected, process flow moves to 1005 where the system, given a user input as to  $\lambda$ , as described above, magnifies or reduces the object or objects, optionally changes the level of detail as described above in the Section entitled "Automatic Activation Of Measures to Preserve Performance", and  
10 automatically moves the objects closer to or farther from the center of the viewing area, as described above. Process flow then passess to 1006.

At 1006, if the size of the magnification factor is such that performance degradation may ensue, the system activates the clipping box so as to preserve a high level of  
15 display performance. Alternatively, if a high magnification value had been in effect previously, and the value of  $\lambda$  is decreased such that the load estimate dips below the applicable threshold value, and there is thus no longer a need for activation of the clipping box, the system will deactivate the clipping box and allow for full viewing of the model by the user. Alternatively, in other exemplary embodiments, other methods  
20 of modifying the load on the system may be applied, as described, for example, in Section 6 above.

At 1007, the system determines whether the user wishes to terminate the zoom operation. If “YES,” process flow moves to 1008, and zoom operation stops. If “NO,” then process flow returns to 1005 and further magnifications and/or reductions, with the appropriate translations of objects, are implemented.

5

Figure 11 depicts an exemplary modular software program of instructions which may be executed by an appropriate data processor, as is known in the art, to implement an preferred exemplary embodiment of the present invention. The exemplary software program may be stored, for example, on a hard drive, flash memory, memory stick, optical storage medium, or other data storage devices as are known or may be known in the art. When the program is accessed by the CPU of an appropriate data processor and run, it performs, according to a preferred exemplary embodiment of the present invention, a method for controlling the scaling of a 3D computer model in a 3D display system. The exemplary software program has four modules, corresponding to four functionalities associated with a preferred exemplary embodiment of the present invention.

The first module is, for example, a Input Data Access Module 1101, which can accept user inputs via a user interface as may be known in the art, such as, for example, a zoom function activation signal, a zoom scaling factor, and current crop box and clipping box settings, all as described above. A second module is, for example, a Magnification Region Generation Module 1102, which, once signalled by the Input Data Access Module that a zoom function has been activated, displays a Magnification Region around the Optimum Viewing Point in the display. If no model(s) are visible within the Magnification Region the module prompts a user to

move model(s) within the Magnification Region. A third module, the Model Zoom Point Selection Module 1103 receives inputs from the Input Data Access Module 1101 and the Magnification Region Generation Module regarding what model(s) are currently located in the Magnification Region, and applies the defined rules, as

5 described above, to select a Model Zoom Point to be used as the center of scaling. A fourth module is, for example, a Scaling and Translation Module 1104, which takes data inputs from, for example, the three other modules 1101, 1102, and 1103, and implements the scaling operation and translates the Model Zoom Point towards or away from the Optimum Viewing Point as determined by defined rules and the value

10 of the scaling factor chosen by a user.

### **Exemplary Zoom Operation**

To illustrate the functionalities available in exemplary embodiments of the present invention, an exemplary zoom operation to view an aneurysm in a brain will be next

15 described with reference to Figs. 12-18. The screen shots were acquired using an exemplary implementation of the present invention on a Dextroscope<sup>TM</sup> 3D data set display system, from Volume Interactions Pte Ltd of Singapore. Exemplary embodiments of the present invention can be implemented on this device. Visible in the figures are a 3D object and a virtual control palette which appears below it.

20

Fig. 12 depicts an exemplary original object from a CT data set, positioned somewhere in 3D space. In the depicted example, a user intends to zoom into a large aneurysm in the CT data set, in this example a bubble-like object (aneurysm) in the vascular system of a brain pointed to by the arrow. Fig. 13 depicts an activation of

25 zoom mode wherein a user can, for example, move a virtual pen to a virtual “zoom

slider” bead. A system can then, for example, automatically select a Zoom Point, here indicated by the four triangle cross, which here appears buried inside the data. The Zoom Point is selected in this exemplary case at the nearest point in data that intersects an Optimum Viewing Point. A Contextual Structure (the circular area  
5 surrounding the four triangle cross) is also displayed to focus a user’s attention on the Zoom Point.

With reference to Fig. 14, since the system did not automatically find the desired point of interest (i.e., the aneurysm), a user needs to refine the selection of Zoom  
10 Point. Thus, a user can, for example, move the object so that the desired part of the object (in this case the aneurysm) coincides with the Zoom Point (which remains at the Optimal Viewing Point). Throughout this operation, the user holds the pen at the bead of the zoom slider, without pressing the virtual button.

15 Fig. 15 depicts how once the Zoom Point coincides with the aneurysm (as here, an exemplary system can adjust the depth of the Zoom Point as a user moves a 3D object towards it) a user can press a button on the zoom slider. The Contextual Structure, being no longer needed, thus disappears.

20 With reference to Fig. 16, as the user drags the bead of the zoom slider, the magnification of the 3D data set around the Zoom Point begins, and with reference to Fig. 17, when a magnification point reaches a certain value, a Zoom Box can, for example, be activated, which can crop the 3D data set to obtain an optimal rendering time. In systems with very high rendering speeds this functionality can be, for  
25 example, implemented at higher magnifications, or not at all. As can be seen with

reference to Figs. 15-18, a zoom slider can display an amount of magnification, which in these figures is displayed behind the virtual pen, but nonetheless, partially visible.

Finally, with reference to Fig. 18, when a desired magnification of the aneurysm in  
5 achieved, for example, a user stops the movement of the zoom slider and can inspect  
the object.

The present invention has been described in connection with exemplary embodiments  
and exemplary preferred embodiments and implementations, as examples only. It  
10 will be understood by those having ordinary skill in the pertinent art that  
modifications to any of the embodiments or preferred embodiments may be easily  
made without materially departing from the scope and spirit of the present invention  
as defined by the appended claims.